

Assessment details for All students

Assessment item 1

Due Date:	Thursday of Week 6 (20th 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, polymorphism, and exception handling. The case study for developing a solution is given below.

Year 11 and 12 students study subjects generally known as Authority subjects or Authority registered subjects. There are many assessments for these subjects in different formats and the grade of these assessments are summative contributing to a student's final result. Entrance to further studies are based on this result and therefore, grades of assessments are critically monitored by students, and their parents. Your task is to develop a **Java Application** for the parents and their year 11/12 kids to monitor academic performance easily. The java application should allow the user to enter student details, and assessments for each subject. Once the assessment results are available the user should be able to set the grade for the assessment including explanatory notes. User should be able to view the subjects enrolled, all assessments for a chosen subject, and all marked assignments with their grades.

A. Graphical User Interface

The application should contain a GUI as shown below. The GUI components should consist of the following panels.

1. A top panel that contains two (2) Text Fields, three Combo Boxes (drop down list), five Labels.
2. A middle panel that contains a Text area to display the Assessment list for a subject or assessment details for a chosen assessment.
3. A bottom panel that contains seven Buttons which are "Create Student", "Load Assessments", "Display Assessments", "Set Grade", "Display Grade" and "Quit".

The functions of the seven buttons are described below.

1. Create Student

Initially the 'Create Student' button remains disabled. The user enters the Student Name and Year Level using the fields on the GUI. This creates a student object. Once the student object is created three core subjects of English, Mathematics B, and Biology, and the chosen elective subjects of, Business and Communication Technologies, and Religion and Ethics are added to the student object. This is done to simplify the process of going through adding each subject.

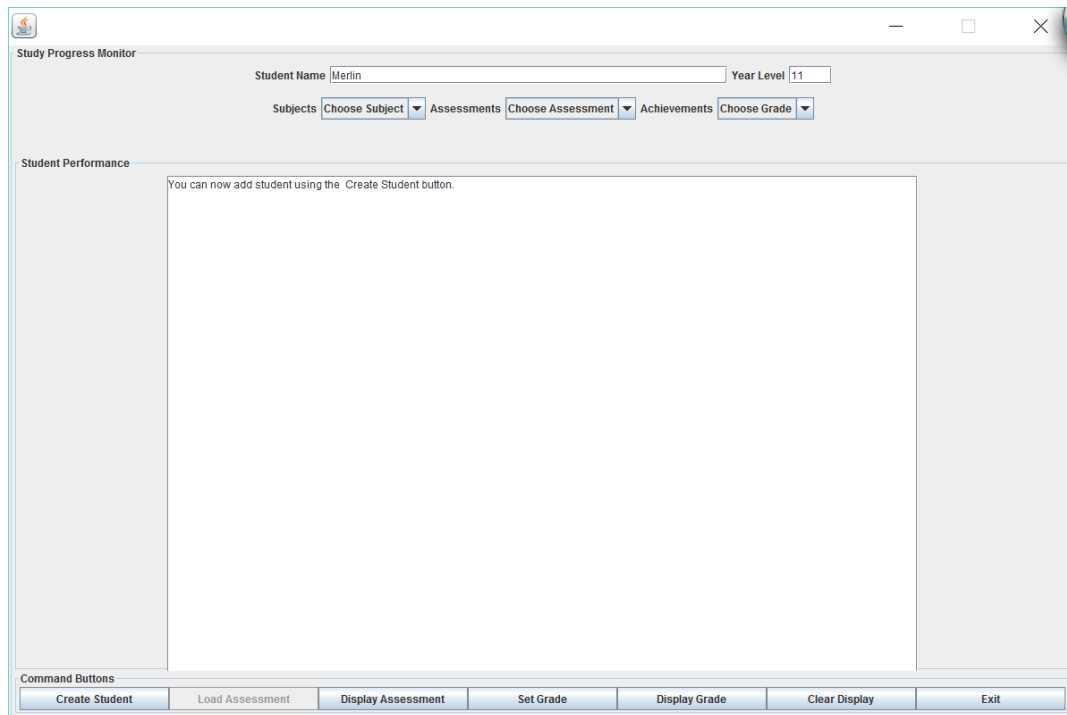


Figure 1 Display after entering Student Name and year Level

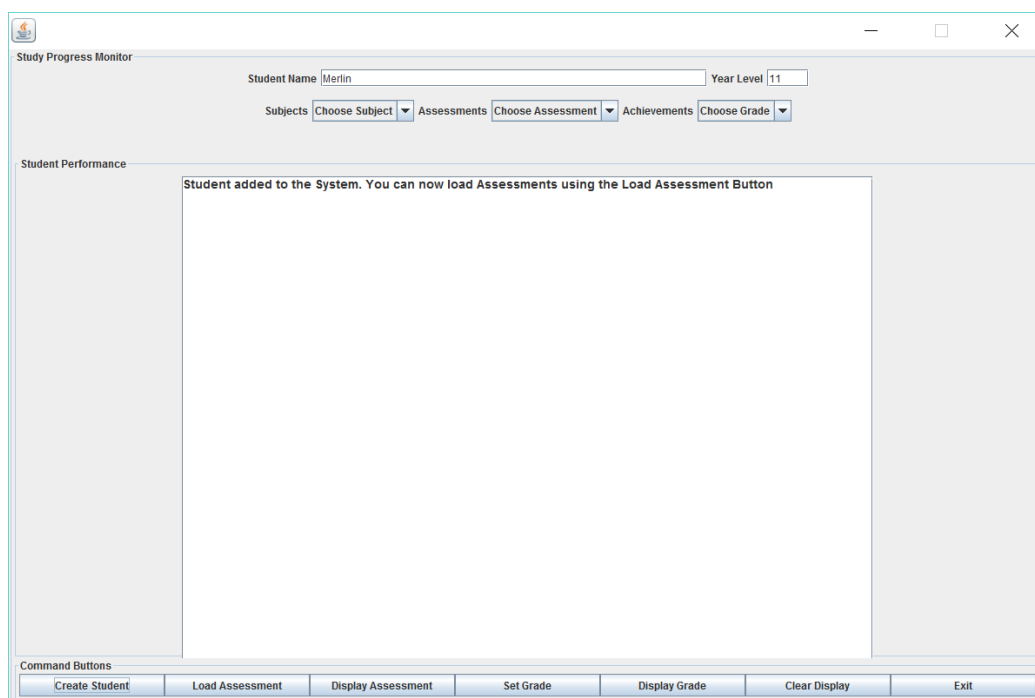


Figure 2 Display after Create Student

2. Load Assessments

Initially the 'Load Assessments' Button remains disabled as shown in Figure 1. This 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 Achievements with corresponding set of values.

Table 1 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

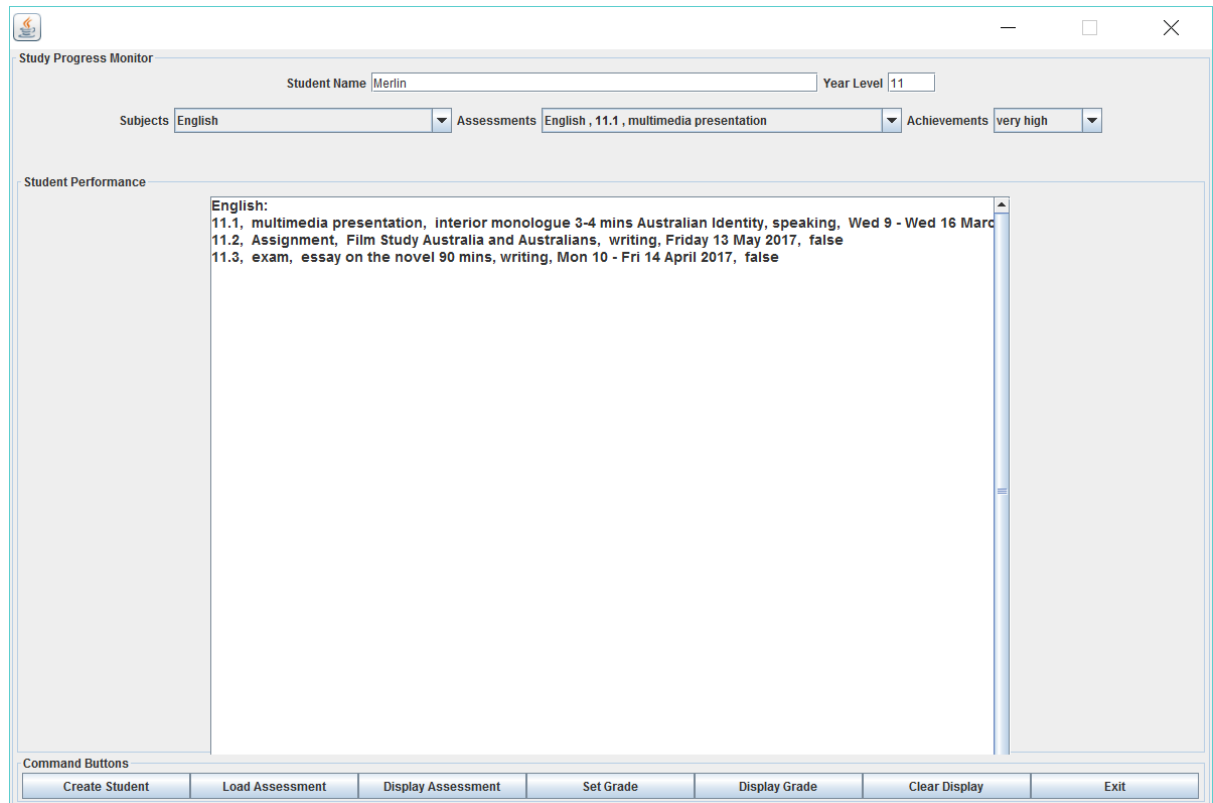
Note: Use the given data file named *COIT20256Ass1Data.csv* available in the Unit website and it contains the set of data required.

The screenshot shows a web application titled "Study Progress Monitor". At the top, there are input fields for "Student Name" (containing "Merlin") and "Year Level" (containing "11"). Below these are three dropdown menus: "Subjects" (selected "English"), "Assessments" (selected "English , 11.1 , multimedia presentation"), and "Achievements" (selected "very high"). The main area is labeled "Student Performance" and contains a message box that says "Student added to the System. You can now load Assessments using the Load Assessment Button". At the bottom, there is a "Command Buttons" section with six buttons: "Create Student", "Load Assessment", "Display Assessment", "Set Grade", "Display Grade", and "Clear Display". The "Load Assessment" button is highlighted, indicating it is the active or next step.

Figure 3: Assessment Loaded

3. Display Assessments

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 a window titled "Study Progress Monitor". At the top, there are input 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 area is labeled "Student Performance" and contains a text box with the following text:

```
English:
11.1, multimedia presentation, interior monologue 3-4 mins Australian Identity, speaking, Wed 9 - Wed 16 Marc
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

4. Set Grade

Clicking on the “Set Grade” button allows the user to enter the Achievements for a chosen Assessment Item. The grading contains levels of achievements, knowledge, and skill. These values are given in the following Table 2. Once the user selects the achievements level using the Combo Box, corresponding levels of knowledge and skill should be added.

Table 2. 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

5. Display Grade

This can be used to display the grades of various assessments set already in the TextArea as shown in Figure 5.

The screenshot shows a Java Swing window titled "Study Progress Monitor". At the top, there are input fields for "Student Name" (containing "Merlin") and "Year Level" (containing "11"). Below these are three dropdown menus: "Subjects" (selected "English"), "Assessments" (selected "English, 11.1, multimedia presentation"), and "Achievements" (selected "very high"). The main area of the window is a large text area labeled "Student Performance" which contains the following text: "English: 11.1, multimedia presentation, interior monologue 3-4 mins Australian Identity, speaking, Wed 9 - Wed", "Achievement: very high", "Knowledge: thorough understanding", and "Skill: uses a high level of skill in both familiar and new situations". At the bottom of the window is a "Command Buttons" bar with seven buttons: "Create Student", "Load Assessment", "Display Assessment", "Set Grade", "Display Grade", "Clear Display", and "Exit".

Figure 5: Display Grade

6. Clear Display

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

7. Exit

This should allow the user to quit the application.

B. Data Structures

The relationships between classes is shown in Figure 6.

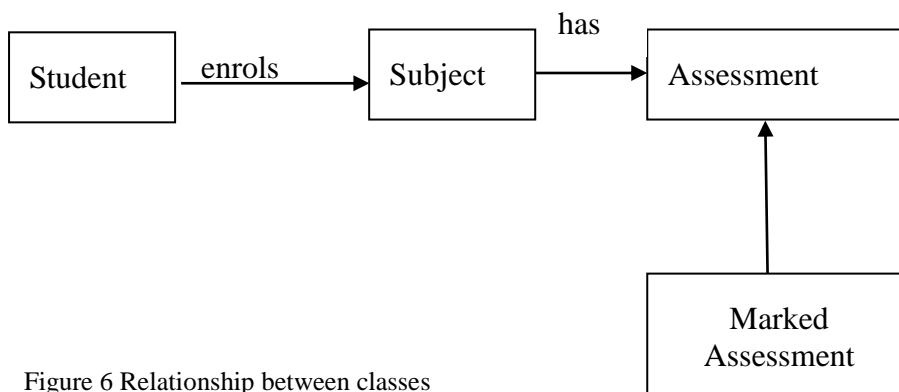


Figure 6 Relationship between classes

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

1. *Student Class*

This is to store the student details of name, year level, and subjects enrolled. A student class can have an ArrayList of Subjects. Use String data type for name and year level.

2. *Subject class*

The purpose of this class is to store the subject name, and the list of Assessments for that subject. Use String data type for subject name, and an ArrayList for Assessments.

3. *Assessment*

Include appropriate fields to store:

Assessment Id, Assessment type, topic, format, dueDate and graded which takes a Boolean value to indicate it is graded or not;

4. *Marked Assessment*

This class extends the Assessment class. This class should have a member fields to store the grading values once the assessment is graded using the Set Grade Button. These values are given in Table 2.

5. *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 a method to read data from the file and load data structures appropriately.
- d) This also contains the main method.

8. 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.)

- Student.java – Source code for the *Student* class as given above
- Subject.java – Source code for the *Subject* class
- Assessment.java – Source code for the *Assessment* class

- MarkedAssessment.java
- StudyMonitorInterface.java – Source code for the GUI components, other required methods as outlined above and the main method.
- Report.docx – File containing UML class diagrams for the classes, Student, Subject, Assessment, and MarkedAssessment. 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)	1.5	
2	Design and use of appropriate data structures	1	
3	Designing classes and using objects, methods, and inheritance	1.5	
4	Use of exception handling (use appropriate exception classes and informative messages)	1	
5	Create Student Button function	1	
6	Load Assessment Button function	2	
	Display Assessment Button function	1.5	
	Set Grade Button function	1.5	
	Display Grade Button function	1	
7	“Clear Display” and “Exit” 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.