XJTLU | Computing

CPT111: Java Programming

Semester 1, 2024-25

Coursework 3: Programming Project – A Simple Quiz System

Read carefully — no dispensation will be given for lack of awareness of the rules

Assignment type	This is a group programming assignment. You are required to work as a group of 4 to 5 students, which will be allocated randomly. (You should be able to find your group members' information from Learning Mall (LM), and should contact the instructors if there is any problem.)
Weighing	Total marks available: **100**, accounting for **30%** of overall module marks.
Release date	**1pm CST Wednesday 6 November 2023**
Due date	**11:59pm CST Sunday 1 December 2024**

Learning outcomes

(A) Understand and appreciate the principles of using object-oriented programming techniques for the construction of professional robust, maintainable programs deployed to meet real world business goals;
(B) Design, write, compile, test, debug and execute object-oriented program using an Integrated Development Environment (IDE);
(C) Effectively develop object-oriented programs as a member or a leader of a software development team with continuous development strategy supported by AI technology;
(D) Implement object-oriented programming to represent, display, and manipulate data while mitigating security risks;
(E) Evaluate legal, social, environmental, ethical, diversity, inclusion, and intellectual property issues related to object oriented program development;
(F) Apply knowledge of engineering management principles, commercial context, project and change management in object-oriented program.

Submission platform	Each group is required to submit an electronic copy of your assignment via LM. You are allowed **ONE** submission only. **It is your responsibility to upload the correct document.**
Late submissions	Penalties will apply for any work submitted after the due date unless you have obtained a formal extension prior to the date for submission. **The penalty applied will be 5% of the available marks** for the assignment **for each working day or part thereof** that the assignment is late. The penalty will be capped at five working days (120 hours) from the assignment deadline. Work submitted after five working days will receive a grade of zero.

Continued on next page

Submission confirmation	**We strongly advise you to double check that you have submitted the correct document / final version of your answer.** **Submission of incorrect file** If you have submitted the incorrect file, you should email the correct file to the instructors **prior to the deadline**. **Submitting the incorrect file can result in failure.**
Special consideration	Requests for an extension due to illness, misadventure, or other extenuating circumstances beyond your control will only be considered via a formal application for special consideration through e-Bridge.
Report format	**ALL** answers **must** be written in English. The report must: • be submitted as a .doc, .docx (do not submit PDF or Apple Pages) • contain headings and subheadings • have 3 cm margins • use a legible font (e.g., Calibri, Arial or Times New Roman) • be presented in 11 point font size with 1.5 line spacing • be paginated
Plagiarism and academic misconduct	It is assumed that you are thoroughly familiar with the policies of XJTLU regarding academic misconduct and plagiarism. Ignorance of the rules is not an acceptable defence against an allegation of academic misconduct. **There are no excuses for engaging in plagiarism.** Assignment answers will be checked for plagiarism. Impermissible similarities between student answers (current and former) can be detected by academic integrity software and by instructor, and will be referred to the School's Examination Officer for investigation. Penalties will follow those of the University's Academic Integrity Policy on e-Bridge and can ranged from capped marks to expulsions from the university. For use of Generative Artificial Intelligence (AI), see below.
Use of Generative Artificial Intelligence (AI)	**Simple Editing Assistance** For this assessment task, you may use standard editing software but not generative AI. You are permitted to use the full capabilities of the standard editing software, such as Microsoft Office suite, Grammarly, and Integrated Development Environments (IDEs) for coding. If the use of generative AI such as ChatGPT is detected, it will be regarded as serious academic misconduct and subject to the penalties mentioned above.
Tips	• Read the questions carefully. • Write succinctly, and avoid repetition. • Avoid being overly descriptive. • Remember to save/back up your work regularly. XJTLU provides all students free access to XJTLU Box. It may be prudent to save your work to your XJTLU account so that you can access it from multiple devices in case you encounter hardware issues. • You are encouraged to post administrative/procedural questions about the assignment on the LM Q&A Forum. The instructors will answer for the benefit of all students.

Coursework 3: Programming Project – A Simple Quiz System

The purpose of this assignment is to design and develop an application that can be used to facilitate educational quizzes, allowing users to select topics of interests, take quizzes related to those topics, and view their quiz scores on a personal dashboard. The primary aim is to create a user-friendly and interactive platform that enhances the learning experience through topic selection and question assignment.

1 Functional requirements

As this is a programming assignment, not a software design one, here is a skeleton to *guide* you in the creative part.

1.1 At startup

- The program should load data, such as user information, questions, etc., from selected files, and store it in memory (using relevant variables and data structure).
  - You may assume that all user information are stored in a comma-separated values (CSV) file[1], as illustrated in `resources‣user.csv`, where the 1st, 2nd, and 3rd columns of the file represent the user id, user name, and password of the user, respectively.
  - You can assume that all questions are stored in some XML files, as illustrated in folder: `resources‣questionsBank`. A Java class `ReadQuestions` has been provided to you in the `src` folder demonstrating how the questions can be read from the XML files into the memory, and the Javadoc of the classes used can be found under the folder `repositories‣javadoc‣xjtlu.cpt111.assignment.quiz`.

    > ***Note:*** You need to add the library:
    > `repositories‣xjtlu‣cpt111‣xjtlu.cpt111.assignment.quiz‣0.0.1‣xjtlu.cpt111.assignment.quiz-0.0.1.jar`
    > to the class path (of your IDE) first before running the `ReadQuestion` class.
- The program should:
  - show information about data loaded.
  - validate the questions read. That is, in our case, a question is valid if it
    (i) falls into a topic (either new or an existing one)
    (ii) has a question statement,
    (iii) has more than one answer available for selection, and
    (iv) has one-and-only-one answer.
  - show a menu that present a list of interactions possible.
- Besides, you may need to add more questions (under different topics) to the question bank yourself.

[1] CSV file: a text file format that uses commas to separate values, and newlines to separate records.

1.2 Menu — General behaviour

The application should support, at least, the following functions.

- User registration and authentication
- Topic selection
  - The topics available should be based on the input from the questions.
- Quiz taking
  - You can decide the number of questions in a quiz. However, each quiz should contains questions at *different* level of difficulties.
  - The questions should be shown to the user one-by-one!
  - The order of answers available for selection should be shuffled every time before a question is shown to the user.
- The score of the quiz should show to the user immediately after the quiz has been finished and the results should be saved into a *score file*
  - You should design the structure (and format) of the score file on your own, and make sure that you can read its content back when the program start!
- User dashboard for viewing quiz results of each topic and history of the previous 3 tests attempted.
- Leaderboard for showing the names of the users with the highest score in each topic.

1.3 Technical requirements

The objective of this assignment is for you to program a simple quiz system so it should be interactive in the form of showing text to the user and requesting their answer and data update through user input as well as reading data from different file types. If you fail to do so, your submission will automatically attract a penalty of **50%**.

1.4 Concept requirements

Your code must feature and make use of the following elements.

- All your code must be divided in relevant/meaningful functions and classes.
- Declare/define and use variables with the appropriate data types and meaningful names.
- Input will always be converted into proper data type.
- The program should prevent ***crash*** in any situation, thus make sure you test your program properly.

1.5 Constraints

Dependencies Using libraries or modules that are **NOT** covered in this course is ***strictly prohibited*** and will result in **zero (0)** marks automatically in this assignment.

Internet connectivity The application is assumed to be run locally on a computer; hence no internet connection is required.

2 Important reminder

Documentation is also a critically important part of your software engineering. Your use of comment (in Java source file) will be graded.

You must write the final version of the program ***totally on your own***. Sophisticated plagiarism detection system are in operation, and they are pretty good at catching copying! Re-read the policy on the university home page, and note the University's tougher policy this year regarding cheating.

Using libraries or modules that are **NOT** covered in this course is ***strictly prohibited*** and will result in **zero (0)** marks automatically in this assignment.

Your programming style (how clearly and how well you speak a programming language, i.e., Java in this course) is what will be graded. Correct functioning of your program is *necessary* but ***not*** *sufficient*; you must use the concepts covered in class and meet all requirements stated in this assignment and as detailed in the marking rubrics (Section 4).

3 Submission requirements

Your assignment has to submit to Learning Mall (LM). You should submit the following:

A. A report (written using Microsoft Word) that includes the following items:
   1. A cover sheet stating the student number of your team members.
   2. A description of your project
   3. Information and explanation of how you add, store and handle users' data (including the data structure that you used to store users' data in memory), the algorithm that you used to identify the user names in the leaderboard, how to handle problems that appear in the questions (e.g., questions with more than one correct answers) and other parts of the program, etc.
   4. Tests performed to verify the correctness of the program.
   5. Printouts/screen capture of your program's execution and tests. (This can be incorporated into Items 3 and 4 above)
   6. Peer evaluation form (Section A). (This will be used to adjust the final mark of individual students according to his/her contributions to the project.)

B. **ALL** Java source codes and resource files (e.g., images, questions and users files, etc.) that are required to run the application.

You should compresses all files into a **single** ".zip" file before submission. You should **NOT** include any files from the `repositories` folder or any files that are **NOT** relevant to the application into the submission file. Failure to follow this requirement will result in mark deduction.

4 Marking

Criteria	Ratings			
	10 to > 8pts	8 to > 5pts	5 to > 0pts	0 pts
Commenting and naming convention (10 pts)	Everything is appropriately commented and every variable is appropriately named	A few comments or variable names are missing or inappropriate	More than a few comments are missing or inappropriate but more than 50% of the elements are commented; OR more than a few variables are inappropriate	There is less than 50% commented OR more than a few variables are not properly named
Prevent crashes (10 pts) Program does not crash in any situation	No program crash	Minor error – program crash for some very uncommon case	Major error – program crash in case of unexpected type of user input OR program crashes frequently	Crashes most of the time (or always) when tested OR circumvent the requirement to handle wrong kind of user input
Appropriate data structure and variables (10 pts) Users and scoring information is stored in memory in a relevant way and with appropriate data types	Users and scoring information are stored in memory in a relevant way and with appropriate data types	Used more than necessary variables OR a few variables are not typed properly	Does not use proper data structure OR data type to most variable (more than once)	NO data structure (only simple data types used) OR more than one error
Loading data (10 pts) Load data in memory and format data in file properly	No issue found when loading data into memory	A small number of data is not loaded (or improperly) or other minor error	One type of data for all items is not loaded or the file is not formated properly or other major error	Data is not loaded or more than one major error
Menu (10 pts) Menu shows the required list of options, and execute the right the functionality based on user input, and show proper message to user in case of any problem	The menu is shown correctly and all options can be selected without any error	Some options are not shown to the user OR some options cannot be selected OR any other minor error	Menu is not shown "infinitely" until the exit option is selected OR any other major error	There is no menu OR it is shown only once OR more than one major error
Show question (10 pts) Show the information about a question and with answers shuffled	The questions are shown according to the topic selected and with different level of difficulties and answers shuffled	The questions are shown according to the topic selected and with answers shuffled, but most questions are of the same level of difficulties	The questions are shown according to the topic selected but without shuffled the answers or different level of difficulties	The questions are shown arbitrary
Scoring (10 pts) Accumulate the score correctly w.r.t. to users' input	The score is updated correctly in all cases	The score is updated correctly in most cases but not in all cases	The score is updated correctly occasionally	The score is updated incorrectly in most cases or not update at all
User dashboard (10 pts) User dashboard shows the user's achievement in each topic and history of the previous 3 tests correctly	All information are shown correctly, especially after scores update	One or two information are shown incorrectly, or with some minor error	Most information are not shown correctly OR with major error	Several major errors OR no information is shown on the user dashboard
Leaderboard (10 pts) Leaderboard shows the name of user with the highest score in each topic correctly	All information are shown correctly, especially after scores update	One or two information are shown incorrectly, or with some minor error	Most information are not shown correctly OR with major error	Several major errors OR no information is shown on the leaderboard
Save data (10 pts) Save data to file with proper formats	All data are saved in a proper format	A small number of data is not saved (or improperly) or other minor error	One type of data for all items are not saved OR file is not formated properly OR it overrides original data OR other major error	Several major errors or there is no saved information in the new file

A Peer evaluation form

Individual contribution for Group Programming Assignment

Name	Student Id	Task(s)	Contribution (%)	Signature