**Computing Department**

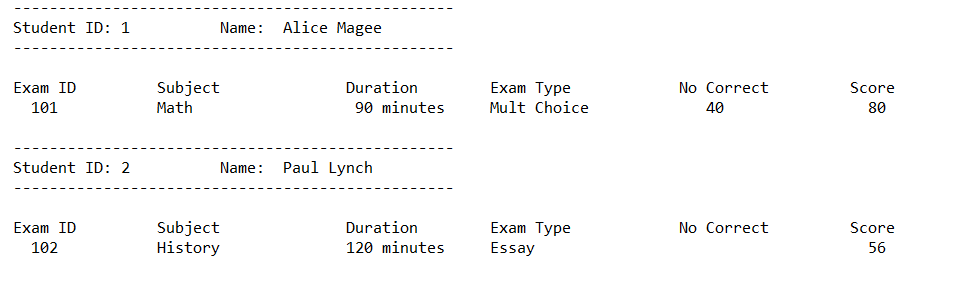
BSc Contemporary Software Development

|  |  |
| --- | --- |
| **Module Name:** | **Software Development** |
| **Assignment No:** | **CA1** |
| **Assignment Title:** | **Design, implement and test a software solution** |
| **Lecturer:** | **Teresa Deeney** |
| **Release Date:** | **13th October 2023** |
| **Submission Date:** | **6rd November 2023** |
| **Indicative Content Covered by Assignment** | **Principles of Object-Oriented Programming**  **Exception Handling**  **Software Testing** |
| **Definition of Plagiarism:** | Plagiarism is the act of taking or copying someone else’s work, including another student’s, and presenting it as if it were one’s own. Plagiarism is said to occur when ideas, texts, theories, data created artistic artifacts or other material are presented without acknowledgement so that the person considering the work is given the impression that what they have before them is the student’s own work when it is not. Plagiarism also occurs when a student’s own work is re-presented without being properly referenced. Plagiarism is a form of cheating and is a disciplinary offence. |
| **Student Declaration:** | I declare that this is my own work and that any material I have referred to has been accurately and consistently referenced. I have read and understand the definition of plagiarism given above. If it is shown that material has been plagiarised, or I have otherwise attempted to obtain an unfair advantage for myself or others, I understand that I may face sanctions. A mark of zero may be awarded and the reason for that mark will be recorded on my file. |
| **Student Name:** | Michael O Brien |
| **Student Signature:** | Michael O Brien |
| **Date Submitted:** | 6/11/23 |
| **Mark:** |  |
| **Assessor:** |  |

**Software Development CA1**

You are tasked with designing a Java program for managing exam results for multiple students in a university. The system should include the following:

1. An interface called ***Scorable*** containing the following method:
   1. *calculateScore()*: This method should be implemented by classes that represent exams and calculates the score of the exam
2. An abstract class called ***Exam*** containing the following attributes:
   1. *examId* (integer) to uniquely identify each exam
   2. *subject* (string) to represent the subject of the exam
   3. *duration* (integer) to store the duration of the exam in minutes
3. Two classes, ***MultipleChoice*** and ***Essay,*** that inherit the ***Exam*** class and implement the ***Scorable*** interface
   1. For *MultipleChoice*, include the following additional attributes:
      1. *correctAnswers* (integer) to store the number of correct answers
      2. *noQuestions* (integer) to store the number of questions on the exam
   2. For *Essay*, include the following attributes:
      1. *essayAnswer* (string) to store the essay answer written by the student
      2. *grammar* (integer) to store the mark attained by the student for grammar
      3. *content* (integer) to store the mark attained by the student for essay content
      4. *wordLimit* (integer) to store the upper word limit for the essay
   3. *Essay* should also contain a method *gradeEssay*() that calculates the essay score (there is no set method, you decide, but it should be based on grammar, content and word count, eg, you could allocate x marks for each, weighting them as you see fit, and penalise if the word count goes a certain percentage above or below the number of words allowed. Use a string method to work out the number of words in the essay answer supplied for this!)
   4. Implement a method *displayExamDetails*() in both *MultipleChoice* and *Essay* classes to display the exam details and results on screen
   5. When implementing the calculateScore() method from the *Scorable* interface for both classes, assume the mark being returned is out of 100, eg, if you have 25 questions and 15 correct answers, the score is 60%.
4. Create an ***ExamException*** class and use it to add the following validation to the ***Exam***, ***Essay*** and ***MultipleChoice*** classes:
   1. *duration* should be between 30 and 180 minutes (Exam)
   2. *wordCount* should be greater than or equal to 0 (Essay)
   3. *wordLimit* should be between 500 and 10000 (Essay)
   4. *noQuestions* should be between 10 and 50 (MultipleChoice)
   5. *correctAnswers* should be greater than or equal to zero (MultipleChoice)
5. Create a class ***Student*** with the following attributes:
   1. *studentId* (integer) to uniquely identify each student
   2. *studentName* (string) to store the name of the student
   3. *examsTaken* (a list of Exam objects) to store the exams taken by the student
6. Create a *StudentException* class and use it in the ***Student*** class to ensure that *studentName* is between 2 and 30 characters in length
7. Create a ***Printable*** interface with methods *printSummaryResult*() and *printDetailedResults().*
8. Implement the ***Printable*** interface in the ***Student*** class to allow printing of student results to a text file (Loads of info available of the internet but you could start here https://www.homeandlearn.co.uk/java/write\_to\_textfile.html). Output should be formatted appropriately. It does not have to look exactly like the example below but it should be clearly laid out in tabular format.



Both print methods should be fully implemented.

1. Create a class ***ExamResult*** that stores the result of an exam for a specific student. It should have the following attributes:
   1. *student* (a Student object) representing the student who took the exam
   2. *exam* (an Exam object) representing the exam taken by the student
   3. *score* (integer) representing the score achieved by the student in the exam
   4. Implement the ***Comparable*** interface in the *ExamResult* class to allow for comparison of results based on scores
2. Create a main tester class ***ExamManagement*** that demonstrates the use of these classes. It should use ArrayLists and create instances of Student, Exam, and ExamResult. It should also contain a menu driven system to allow exam results for multiple students to be entered from the keyboard and stored. These results should then be displayed (offer several options here, eg, by exam type, sorted by score, etc) and a print to file option offered if required.