Assessment Criteria:		Total Marks	Marks
1.	Program Execution, Completeness & Correctness	**5**	
2.	OOP Implementation	**5**	
3.	UI Design (console based interface)	**2**	
4.	Business Rules Implementation	**2**	
5.	Input Validation (try-catch)	**1**	
6.	File & I/O Handling	**1**	
7.	Program Structure and Readability (e.g. alignment, naming convention, commenting code)	**1**	

Assignment Overview

Marking and grading are tedious processes in the education context. This assignment is required your team to create a prototype of a Lab Test System to automate the examination process to increase the productivity of teaching staff in the education industry.

Task Specifications:

- Create the following classes to serve as a model layer in your program.

  - **Question** class. It is a class to hold the examination question details in multiple-choice question mode. This class should consist of the following details:
    - Question text (e.g. `Integer data type holding how many bytes in memory?`)
    - Options from A to D (e.g. `A-2, B-3, C-4, D-8`)
    - Actual answer (e.g. `C`)

    Provide basic functionalities for this class.

  - **QuestionBank** class. It is a class to hold all the MCQ questions loaded from the file. The following class diagram shows the basic structure of this class:

QuestionBank
- questions: Question[] `//a set of question` - title : String `//question bank title such as Java Programming`
+ QuestionBank() `// constructor` + loadQuestions(string file): void `//load all questions from text file` + getQuestions(int) : Question[] `//return a set of random questions according to the size of parameter` + showQuestionsDetails(): void `//display question bank details such of how many questions in the question bank` + resetQuestionBank: void `//clear all the preloaded questions` + viewQuestions() : `show 3 questions in page by page basic` + getQuestionBankSize(): `return total question in the question bank` `+ other getter and setter methods`

- Create a driver program called **Examination** which allows the user to perform the test. Here is the simple menu interface:

```
Examination



   1) Start Examination
   2) Export Results
   3) Exit
```

- `Start Examination` option allows the candidate to start the test. The candidate is required to enter his/her id and exam date before the exam starts. You should provide basic validation on candidate answers (only. A, B, C or D are accepted). For example:

```
Enter your student ID: P10001
Question 1/10

How many bytes memory for int data type?
A. 1       B.  2
C. 3       D.  4

Enter your answer: _
```

  After the test, the system will display the test statistics such as total correct, total incorrect and overall mark in 100% mode.

- `Export Results` option allows the examiner to export all the test results to a text file with the name of student id (`P10001.txt`) as the following format:

```
Examination: Java Programming Lab Test  12/11/20221
Student Answers:
P10001
 1) a (c)
 2) d
 3) b
 4) a
 5) d (a)
 6) a (b)
 7) a (b)
 8) c
 9) d
10) b (c)

(P10001): 50% E
```

Show the score both as a percent of the maximum score, and as a letter grade, according to the following chart:

- A: 90% or greater
- B: 80% – 89%
- C: 70% – 79%
  D: 60% – 69%
- E: 59% or lower