Your organization needs an Examination system.

Design a Class to represent the Question Object, Question is constructed from a Body, Marks, Header and question type.

We want the application to accept different Question Types. True or False, choose one and Choose all. each has a different way of representation.

We need to define a Base Question class and every type as an inherited one.

**Design a class to represent the Question list by inheriting the List<> class**

**Override the Add Method, keep the default behavior for the Add Method and add logic to open a file and Log the Questions in it, every Question Object of Question List will be logged to a Same file. each Question List has Different File**

**(help on TextWrite and TextReader Class ).**

We need to define a class for the answers and also the Answer List.

Question Object is associated with an Object of AnswerList

Design a Base class: Exam, exam class describe the common attributes concerning the exam, StartTime, EndTime and number of Questions, **Question Answer Dictionary(Which will be used for Exam Correction)** , a Show Exam Functionality that it’s implementations will be differed to the further classes in the hierarchy

Every Exam object is Associated to a Subject Object (implement any desired Subject class members)

We have two types of Exams , Practical Exam and Final Exam , Practical exam shows the right answer after finishing taking the Exam , while the Final Exam Only Shows The Question and Answers .

**You want to consider what type of constraints you need to add to this Generic class**

In the Main method declare two objects one of practice exam and one final Exam

We need the end user to select the Exam Type , and upon this choice we will show the Exam .

Implement Iclonable , IComparable , consider overriding ToString , **Equals , GetHashCode** all the constructors use chaining .

**Every exam has a mode : Starting , Queued , Finished**

**When the exam in Starting Mode , Every Student taking this subject should be notified ( implement the desired class hierarchy and implement the required evetns and delegate to produce this functionality )**

تحتاج منظمتك إلى نظام امتحانات. قم بتصميم فئة لتمثيل كائن السؤال، حيث يتكون السؤال من نص، وعلامات، ورأس. نريد من التطبيق قبول أنواع مختلفة من الأسئلة، مثل صحيح أو خطأ، اختر واحدًا، واختر الكل، حيث لكل منها طريقة تمثيل مختلفة. نحتاج إلى تعريف فئة سؤال أساسية وكل نوع كفئة وراثية. قم بتصميم فئة لتمثيل قائمة الأسئلة من خلال وراثة فئة List<>. قم بتجاوز طريقة الإضافة، مع الاحتفاظ بالسلوك الافتراضي لطريقة الإضافة، وأضف منطقًا لفتح ملف وتسجيل الأسئلة فيه، حيث سيتم تسجيل كل كائن سؤال من قائمة الأسئلة في نفس الملف. كل قائمة أسئلة لها ملف مختلف (مساعدة في فئة TextWriter وTextReader). نحتاج إلى تعريف فئة للإجابات وأيضًا قائمة الإجابات. كائن السؤال مرتبط بكائن قائمة الإجابات. قم بتصميم فئة أساسية: امتحان، تصف فئة الامتحان الخصائص المشتركة المتعلقة بالامتحان، مثل الوقت، عدد الأسئلة، قاموس سؤال-إجابة (والذي سيستخدم لتصحيح الامتحان)، ووظيفة عرض الامتحان التي ستختلف تنفيذاتها في الفئات الفرعية في الهيكل الهرمي. كل كائن امتحان مرتبط بكائن موضوع (قم بتنفيذ أي أعضاء موضوع مرغوب فيها). لدينا نوعان من الامتحانات، امتحان تدريبي وامتحان نهائي، حيث يظهر الامتحان التدريبي الإجابة الصحيحة بعد الانتهاء من الامتحان، بينما يظهر الامتحان النهائي فقط السؤال والإجابات. تريد أن تأخذ في الاعتبار نوع القيود التي تحتاج إلى إضافتها إلى هذه الفئة العامة. في الطريقة الرئيسية، قم بإعلان كائنين، واحد لامتحان تدريبي وآخر لامتحان نهائي. نحتاج إلى أن يختار المستخدم النهائي نوع الامتحان، وعند هذه الاختيار سنعرض الامتحان. قم بتنفيذ ICloneable وIComparable، واعتبر تجاوز ToString وEquals وGetHashCode، وجميع البناة يستخدمون السلسلة. كل امتحان له وضع: بدء، في قائمة الانتظار، منتهي.

عند بدء الامتحان، يجب إبلاغ كل طالب يأخذ هذه المادة (قم بتنفيذ التسلسل الهرمي المطلوب للفئات وتنفيذ الأحداث المطلوبة وتفويضها لإنتاج هذه الوظيفة).