**1. Classes:**

a. **EngineeringEducatorQuestionHandler** : Will have responsibility of selecting questions, having a count of total number of questions and making sure unique questions are selected on each run.

b. **EngineeringEducatorQuestion**: Will have responsibility of each question. Question comprises of : Model, FBD and assumptions. So this class will have information of these three attributes only. Main functionality:

* Reading Model and fbd images
* Reading Questions.txt
* Storing assumptions as list of objects (Hashmap is used, key = assumption, value = obj of that assumption)
* Creating check box list of assumptions
* Reading Max score, perAssumptionScore, perReasonScore (can be transferred to Assumption class since Question has nothing to do with reasons)
* Score calculation based on Assumption selection

c. **EngineeringEducatorAssumption:** Will have responsibility of each assumption (assumption string and corresponding answer are initialized in constructor). Incorrect/complicated assumption comprises of set of reasons. So this class will have information of list of reasons. Main functionality:

* Storing reasons as list of objects (Hashmap is used, key = reason, value = obj of that reason)
* Creating radio button list of reasons
* Creating customized checkbox
* Score calculation based on Reason selected

d. **EngineeringEducatorReason:** Will have responsibility of each reason (reason string and corresponding answer are initialized in constructor). Main functionality:

* Creating customized radio button

e. **EngineeringEducatorMainPage:** Will have responsibility of displaying stuffs on GUI. It will retrieve images and list of assumptions from Question class; list of reasons from Assumption class. So it should have only two class objects.

GUI is not responsible for question folder selection. Prof said GUI should do only one thing : display stuffs. We will have to find some way to modify this class such that it is not dependent upon QuestionHandler class.

**2. Question text file format:**

Name: Questions.txt

Top 3 lines will contain MaxScore, ScorePerAssum (will be helpful while calculating partial score), ScorePerReason(will decide how much score to give for valid reason)

Each incorrect/complicated assumption are in following format:

**Assumption String|incorrect(or complicated)**

**Incorrect(or Complicating) Assumption #<Some number>**

**List of reasons #<Some hash>|Corresponding answer (valid or invalid)**

**End Reasons**

Refer text file for more details